PTO 08-6267

CC=JP
DATE=20021206
KIND=A
PN=02353998

COMMUNICATION DEVICE AND NETWORK SYSTEM EMPLOYING SAME AND A SPANNING TREE BUILDUP METHOD

[TSUSHIN SOCHI OYOBI SORE O MOCHIITA NETOWAAKU SHISUTEMU OYOBI NI SUPANINGU KOCHIKU HOHO]

YASUAKI SHIBAZAKI

UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. JULY 2008
TRANSLATED BY: SCHREIBER TRANSLATION,INC.

PUBLICATION COUNTRY	(10):	JP
DOCUMENT NUMBER	(11):	02353998
DOCUMENT KIND	(12):	A
PUBLICATION DATE	(43):	20021206
APPLICATION NUMBER	(21):	2001-161541
APPLICATION DATE	(22):	20010630
INTERNATIONAL CLASSIFICATION	(51):	H 0 4L 12/46
PRIORITY COUNTRY	(33):	NA
PRIORITY NUMBER	(31):	NA
PRIORITY DATE	(32):	NA
INVENTOR(S)	(72):	SHIBAZAKI, YASUAKI
APPLICANT(S)	(71):	NEC CORP.
DESIGNATED CONTRACTING STATES	(81):	NA

TITLE (54): COMMUNICATION DEVICE AND NETWORK SYSTEM EMPLOYING SAME AND A SPANNING TREE BUILDUP METHOD

FOREIGN TITLE [54A]: TSUSHIN SOCHI OYOBI SORE O MOCHIITA NETOWAAKU SHISUTEMU OYOBI NI SUPANINGU KOCHIKU HOHO

[Claim]

[Claim 1] A communication device which is used to interconnect a plurality of networks, the invention characterized as having a plurality of spanning tree management means used to manage spanning trees inside the abovementioned plurality of networks.

[Claim 2] A communication device as described in Claim 1 wherein each of the abovementioned spanning tree management means is such that bridge IDs which are identification information used to manage the spanning tree are provided in advance and the spanning tree construction processing procedures are executed using this bridge ID.

[Claim 3] A communication device as described in Claim 1 or Claim 2 having a control means used to control by allocating the abovementioned plurality of spanning tree management means for a plurality of ports which are interfaces with each of the abovementioned network.

[Claim 4] A communication device as described in Claim 3 wherein the abovementioned control means carries out control and allocation as indicated above by referencing tables whose allocation status with the abovementioned

plurality of ports and the abovementioned plurality of spanning tree management means have be stored in advance.

[Claim 5] A communication device as described in Claim 4 wherein each of the abovementioned spanning tree management means executes the abovementioned spanning tree configuration processing procedures through an exchange with a bridge protocol data unit (BPDU) and another communication device, the abovementioned control means determines to which spanning tree management means the port receiving the abovementioned bridge protocol data unit is to be allocated by referencing the abovementioned table.

[Claim 6] A network system comprising a plurality of networks and a communication device which interconnects these networks, the abovementioned communication system characterized as having a plurality of spanning tree bridge management means used to manage the spanning trees inside the abovementioned plurality of networks.

[Claim 7] A network system as described in Claim 6 or Claim 7 wherein each of the abovementioned spanning management means is such that bridge IDs, which are identification information used to manage the spanning tree are provided separately beforehand and which execute the spanning tree configuration processing procedures using this bridge ID.

[Claim 8] A network system as described in Claim 6 or Claim 7 comprising a control means used to control by allocating the abovementioned plurality of spanning tree management means.

[Claim 9] A network system as described in Claim 8 wherein the abovementioned control means controls by allocation as indicated above by referencing tables in which the allocation condition with the abovementioned plurality of ports and the abovementioned plurality of spanning tree management means is stored in advance.

[Claim 10] A network system as described in Claim 9 wherein each of the abovementioned spanning tree management tree means executes the abovementioned spanning tree configuration processing procedures through an exchange between the bridge protocol data unit (BPDU) and another communication device.

[Claim 11] A method of constructing a spanning tree in a network system comprising a plurality of networks and a communication device having plurality of spanning tree management means used to interconnect these networks and to manage the spanning tree, the invention characterized as comprising steps used to execute spanning tree configuration processing procedures using a bridge ID, which consists of identification information provided

beforehand for the abovementioned spanning management in each of the abovementioned spanning bridge management means.

[Claim 12] A spanning bridge configuration method as described in Claim 11 wherein the abovementioned spanning tree buildup processing procedures are executed through an exchange with the bridge protocol data unit (BPDU).

[Claim 13] A spanning tree buildup method as described in Claim 12 comprising a step wherein a determination is made as to which spanning tree management means the port which has received this bridge protocol data unit is allocated in response to receiving the abovementioned bridge protocol data unit by referencing a table stipulated in advance.

[Claim 14] A program used to execute by computer the control of operations for the communication device to interconnect the plurality of networks, the invention characterized as comprising a control step wherein a plurality of spanning tree buildup processing procedures are controlled and allocated to the plurality of ports which are the interface for each of the abovementioned networks.

[Claim 15] A program as described in Claim 14 wherein the abovementioned control step controls and allocates in a state wherein the abovementioned plurality of ports and the

abovementioned plurality of spanning tree buildup processing means are allocated by referencing a table which has been stored in advance.

[Claim 16] A program as described in Claim 15 wherein each of the abovementioned spanning tree buildup processing procedures are executed by an exchange between the bridge protocol data unit (BPDU) and another communication system; in the abovementioned control step, a determination is made as to which spanning tree buildup processing procedure the port receiving the abovementioned bridge protocol data unit is to be allocated by referencing the abovementioned table.

[Detailed Description of Invention]

[0001]

[Technical Field] The present invention relates to a method for buildup of a communication device, a network system using this and a spanning tree, and particularly a spanning buildup method in a network system wherein a plurality of

/3

networks are interconnected by a bridge.

[0002]

[Description of the Prior Art] We shall explain the bridge network using the prior-art spanning tree referring to

Figure 7 and Figure 8. In the network indicated in Figure 7, bridge 1, bridge 3 and bridge 4 are connected respectively by LAN A. In addition, bridge 1, bridge 2 and bridge 5 are interconnected respectively by LAN B. In addition, bridge 2, bridge 3 and bridge 4 are interconnected by LAN C.

[0003] In addition, bridge 5, bridge 6, bridge 8 and bridge 9 are interconnected respectively by LAN D. In addition, bridge 7, bridge 8 and bridge 9 are respectively interconnected by LAN E. Bridge 6 and bridge 7 are interconnected by LAN F.

[0004] Each bridge from bridge 1 through bridge 9 is provided with a function which builds up the spanning tree through an exchange with configuration bridge protocol data unit (C-BPDU) stipulated in IEEE802.1d. Figure 10 indicates the condition of the spanning tree which has finally been built up through the exchange with CBPDU

[0005] In Figure 10, bridge 1 is determined by a root bridge and the spanning tree which has been built up is indicated by the dotted line. By building up the spanning tree in this way, a data packet can be transferred without

generating a loop on a plurality of LAN networks connected by the bridge.

[0006] Furthermore, in Figure 7 and Figure 10, "bridge ID" is an identification number used to identify the bridge; while "port ID" is an identification number used for each port. In addition, "root path cost" and "path cost" are values indicating the ease with which these can arrive at the root bridge. The spanning tree is built up and the value of the path cost set on the bridge (port) each time it passes through the bridge is added so that the this root cost and path cost are used as parameters so that a bridge can be selected where the root path cost is at its lowest (easy to reach).

[0007]

[Problems Which the Present Invention is Intended to Solve]

However, when the prior-art spanning tree was used, there were the following problems. When a link was lost between certain bridges due to a breakdown in the network connecting the bridges, the spanning tree was reconfigured according to the stipulations of IEEE 802.1d. All of the spanning tree configuration information built thus far and the filtering data bases studied for forwarding of the data

packet during reconfiguration were initialized and a new tree was built.

[0008] A considerable amount of time was required depending on the number of bridges making up the reconfigured network for the tree, the data packets to be sent were not transferred between the networks until the reconfiguration was complete and the networks entered a state which was tantamount to an interruption of service.

[0009] It is an object of the present invention to provide a communication device, a network system using this and a spanning tree buildup system and a method for same such that when there is a fault in the network and its component devices, the fault can be restored quickly, minimum parameters for reconfiguring the tree are maintained and service interruptions can be kept to an absolute minimum.

[0010]

[Means Used to Solve the Problems] The communication device in the present invention is used to interconnect a plurality of networks and is characterized as having a plurality of spanning tree management means used to manage the spanning tree inside the abovementioned plurality of

networks. Thus, each of the abovementioned spanning tree management means is such that a bridge ID which is identification information used to manage the spanning tree is provided separately in advance and the spanning tree buildup processing procedures are executed using this bridge ID; and it comprises a control means used to control and allocate the abovementioned plurality of spanning tree management means for the plurality of ports which are the interface for each of the abovementioned networks.

[0011] Then, the abovementioned control means is characteristic in that it controls by allocation by referencing the table in which the allocation state for the abovementioned plurality of ports and the abovementioned plurality of spanning tree management means. In addition, each of the abovementioned spanning tree management means executes the abovementioned spanning tree buildup processing procedures through an exchange between the bridge protocol data unit (BPDU) and another communication device. The abovementioned control device is characteristic in that it determines to which spanning tree management means the port which has received the abovementioned bridge protocol data unit is to be allocated by referencing the abovementioned table.

[0012] The network system in the present invention is a network system comprising a plurality of networks and a communication device which interconnects these networks. The abovementioned communication device has a plurality of spanning tree management means used to manage the spanning trees inside the abovementioned plurality of networks.

[0013] The spanning buildup method in the present invention is a method in a network system comprising a plurality of networks and a plurality of spanning tree management means used to interconnect these networks and to manage the

/4

spanning tree. It is characterized as comprising a step used for execution of spanning tree buildup processing procedures using a bridge ID which is identification information which is provided separately in advance to manage the abovementioned spanning tree in each of the abovementioned spanning tree management means.

[0014] Then, the abovementioned spanning tree buildup processing procedures are characterized as being executed by an exchange with the bridge protocol data unit (BPDU) and another communication device. It is also characterized

as comprising a step which determines to which spanning tree management means the port which has received this bridge protocol data unit is to be allocated in response to receiving the abovementioned bridge protocol data unit.

[0015] The program in the present invention is used to execute to a computer the control of the operations for the communication device used to interconnect the plurality of networks, the invention characterized as comprising a control step which allocates and controls a plurality of spanning tree buildup processing procedures for a plurality of ports which are the interface with each of the abovementioned networks. Then, the abovementioned control step is characterized as carrying out the abovementioned allocation and control by referencing a table in which the allocation state for the abovementioned plurality of ports and the abovementioned plurality of spanning tree buildup processing procedures has been stored in advance. In addition, each of the abovementioned spanning tree buildup processing procedures is executed by an exchange between the bridge protocol data unit (BPDU) and another communication device. The abovementioned control step is characterized as determining to which spanning tree buildup processing procedure the port which has received the

abovementioned bridge protocol data unit is allocated by referencing the abovementioned table.

[0016] We shall now describe the actions of the present invention. It is configured so that the spanning tree is partitioned without partitioning a LAN segment by defining a plurality of spanning trees inside a single bridge in a bridge LAN configured of a plurality of LANs. As a result, a plurality of management parts used to manage the spanning tree are placed inside as a single bridge. A plurality of management parts can be controlled by flexible allocation for a plurality of ports which are the interface with each of the LANs. It is also configured by providing independent bridge IDs for each of these management parts to manage the spanning tree. This makes it possible for each of the management parts to carry out buildup processing for the spanning tree independently. A plurality of spanning trees can be partitioned flexibly inside the same LAN segment and a bridge LAN with a plurality of spanning trees present inside the same LAN segment can be configured.

[0017]

[Mode of Working the Invention] Next, we shall explain a practical embodiment of the present invention referring to

figures. Figure 1 is a schematic conceptual view of the spanning bridge in the practical embodiment of the present invention. In Figure 1, the bridge has a plurality of spanning management parts which can manage a single spanning tree. It is made so that the ports to be managed (made for interface with LAN) by each spanning tree management part 31 - 3 (m is an integer of 2 or greater) are allocated respectively by setting from the outside (network manager and the like).

[0018] [blank] [sic]

[0019] A port-to-spanning tree management allocation table 22 is provided for this and the allocation state is set beforehand from the outside by the network manager or others when the system is designed. CPU (control part) 2 determines to which spanning tree management part the C-BPDU (abbreviated to CBPDU) received from a plurality of ports # 1 through #n is to be distributed by referencing this table 22 and the C-BPDU received is supplied to the spanning tree management part determined.

[0020] Furthermore, frame processing parts 11 through 1n used to process sending data frames and sending C-BPDU to

receiving data frames and receiving C-BPDU from each of the ports are placed between each of the ports and CPU 21. Figure 2 is an example of the contents of the port-to-spanning tree management part allocation table 2. In this example, port #1 is allocated to spanning tree management part #1. It also indicates the state where port #2 and port #3 are allocated to spanning tree management part #3.

[0021] Figure 3 is a block diagram of the configuration of frame processing part 11. It is exactly the same as the configuration of other frame processing part 12 to 1n. Furthermore, in Figure 3, frame processing part 11 is indicated as being allocated to spanning tree management part 31 by CPU 21. CPU 21 is not indicated in the figure.

[0022] In Figure 3, frame receiving part 51 receives a frame from the port. If this is a regular frame, it is supplied to frame transfer management part 53. If is C-BPDU, it is supplied to C-BPDU terminal part 52. Frame transfer management part 53 is used to reference the details of port information table 54 for the receiving frame from frame receiving part 51 and to control whether to relay the frame to frame sending part 55 or to discard it passing through frame sending. Port information table 54

is used to store whether the port state of port #1 is an alternate port or a designated port.

[0023] Frame sending part 55 sends a data frame relayed from frame transfer management part 53 and C-BPDU from C-BPDU generation part 56 to port #1. Port ID/port path cost table 57 is used to store the ID in said port #1 and the path cost, which is the parameter. C-BPDU terminal part 52 receives C-BPDU from frame receiving part 51 and carries out terminal processing. C-BPDU which has been made a terminal is distributed by CPU 31 (see Figure 1) to spanning tree management part 31 which is allocated to port #1. C-BPDU generation part 56 generates C-BPDU upon instructions from spanning tree management part 31.

[0024] Spanning tree management part 31 (like the other spanning tree management part) executes the spanning tree protocol buildup processing procedures separately using C-BPDU which has been made a terminal by C-BPDU terminal part 52. As a result, the abovementioned procedures are carried out by referencing table 58 used to store bridge ID/root cost and port ID/port path cost table 57.

[0025] Figure 4 is an example of the C-BPDU used for spanning building. [BPDU TYPE] indicates that this unit is C-BPDU. The receiving frame is confirmed as being C-BPDU by this and the receiving frame is outputted to the next-level C-BPDU terminal part 52.

[0026] [Root ID] is the bridge number of the bridge which is the root bridge in the spanning tree. [Root path cost] is a value which indicates the ease of reaching the root bridge, as indicated above. [Bridge ID] is the number provided the bridge. [Port ID] is the port number (#). Furthermore, other parameters are not necessarily related to the present invention so that we shall not explain them. However, details of this format are stipulated in IEEE 802.1d.

[0027] Each spanning tree management part 31 through 3m manages the state of the port belonging to each management part based on IEEE 802.1d in accordance with the C-BPDU received by the port to be managed which has been allocated and carries out spanning tree buildup inside each LAN connected to these ports. Spanning tree management parts 31 through 3m are operated respectively independently and the other spanning tree management part is not affected.

[0028] Thus, in the present invention, a plurality of spanning trees are managed independently inside a single bridge. As a result, a spanning tree can be partitioned into a plurality of partitions inside the same LAN segment. A bridge LAN where a plurality of spanning trees exist inside the same LAN segment can be configured.

[0029] Here, Figure 7 indicates the network which is connected to a plurality of LANs inside the bridge. In Figure 7, bridge 1, bridge 3 and bridge 4 are respectively interconnected and bridge 1, bridge 2, bridge 5 are respectively interconnected by LAN B.

[0030] Bridge 2, bridge 3 and bridge 4 are respectively interconnected by LAN C. Bridge 5, bridge 6, bridge 8 and bridge 9 are respectively interconnected by LAN D. In addition, bridge 7, bridge 8, bridge 9 are respectively interconnected by LAN E. Bridge 6 and bridge 7 are respectively interconnected by LAN F.

[0031] Bridges 1 through 9 are such that a bridge ID, a port ID, a path cost) (including the route path cost) are provided by the network manager and others. Examples of the

respective values are indicated in Figure 7. Here, bridge 5 indicates the bridge which loads a plurality of spanning tree management parts 31 to 3m indicated in Figure 1. In the example in the figure, settings are made by the network manager so that the port which is oriented to the top part of bridge 5 is managed by a single spanning tree management part. The port which faces downward is managed by one other spanning tree management part. As a result, two bridge IDs are defined independently on bridge 5 to manage each of these spanning trees.

[0032] In the network indicated in this Figure 7 in the present invention, a plurality of spanning trees are built. In Figure 8, the two spanning tree segments (indicated by the dotted line in 1 and 2) are built by bridge 5. Figure 5 indicates the schematic flow of these buildup procedures. When each of the bridges is operated, sending and receiving of C-BPDU starts for the spanning tree buildup. The root bridge ID, root path cost, bridge ID and other types of information are contained in this C-BPDU. The spanning tree buildup starts with this.

[0033] At this time, when C-BPDU C-BPDU is received (Step 100), CPU21 determines the spanning tree management part by

referencing allocation table 22 based on the port received by C-BPDU (step 101). C-BPDU is distributed to the spanning tree management part determined (Step 102) and the spanning tree buildup processing procedures, which is spanning tree management processing, is executed inside this spanning tree management part (Step 103).

[0034] These spanning tree building processing procedures are executed according to the detailed flow made up of Steps S1 through S15 indicated in Figure 6, based on the stipulations of IEEE 802.1d, as indicated above. However, as these procedures are well known, we shall provide only a summary of them. First, a root bridge is not determined for each bridge so that each of these starts operating as a root bridge. The C-BPDU is sent with a root path cost of 0. At this time, C-BPDU is sent on bridge 5 (see Figure 7) at

/6

values such as the following: root bridge ID = bridge ID = 97 is sent by C-BPDU to the port on the port ID 1 side and root bridge ID = bridge ID = 70 is sent to the port on the port ID 2 side.

[0035] Then, the following determination is made by using the C-BPDU (Step 1) received from each port on each bridge

as well as its own bridge ID. When the root bridge ID of the C-BPDU of any port received is greater than its own bridge ID (step S2), it becomes a root bridge (Step 23).

[0036] When this is not the root bridge (step S4), a port receiving C-BPDU which satisfies each of the following conditions is determined as a root port (step S10).

· the root bridge ID received is smaller,
· when the root bridge ID is the same, the root path cost is smaller,
· when the root ID and the root path cost are the same, the bridge ID is smaller.

[0037] Then, the bridge corresponding to the route bridge ID of the C-BPDU received is used as the route bridge. The route bridge ID of the C-BPDU sent is updated to this value. The value found by adding the path cost which is set inside it to the route path cost contained in the same C-BPDU is updated as the new route path cost value and a new C-BPDU is sent to ports other than the route port.

[0038] These operations are repeated and last of all, the spanning tree is built up by determining the port which continues to send C-BPDU as the designated port (Step S14)

and by determining ports other than the root port which continues to receive C-BPDU as the alternate port (Step S15).

[0039] Through these operations, the root bridge IDs of all of the C-BPDU received on bridge 1 are greater than the bridge IDs set on bridge 1 so that bridge 1 is determined to be the root bridge. On bridges 2 through 5, the root bridge ID contained in the C-BPDU received from bridge 1 is the smallest so that the ports (port ID 1 of each of the bridges) facing bridge 1 are respectively determined to be route ports, as indicated in Figure 8.

[0040] Port 2 of bridge 4 having the smallest root path cost is the designated port so that it can continue to send C-BPDU to the very end between bridges 2, 3 and 4 which are connected in LAN C. Conversely, port 2 of bridges 2 and 3 is the alternate port so that it can continue to send B-BPDU.

[0041] On bridge 5, C-BPDU which has been received at port 1 is such that a single spanning management part is managed inside bridge 5. Port 2, which is defined in a separate spanning tree management part inside bridge 5, is not

affected. As a result, the spanning tree having bridge 1 as a root bridge is terminated only on the port 1 side of bridge 5. However, the exchange with C-BPDU explained previously is also carried out inside the network configured of bridges 6 through 9 which are connected to port 2 of bridge 5. The example in Figure 7 indicates the configuration of the spanning tree having bridge 7 as the root bridge (results of the spanning tree buildup are indicated in Figure 8 by the dotted line).

[0042] Even in this case, spanning tree information from C-BPDU on the port 2 side of bridge 5 does not affect the port 1 side so that the spanning tree having bridge 7 as a root bridge is terminated

[0043] Likewise, a separate spanning tree protocol operates between bridges 5 through 9. The root bridge is determined in bridges 5 through 9 (here, indicated a bridge 6), and a separate spanning tree is built up having that root bridge as its origin. This makes it possible for a plurality of spanning trees to be configured inside the same segment.

[0044] An example of application for a ring network is indicated in another practical embodiment of the present

invention. In the prior art, a bridged LAN was configured by closing it within a geographically narrow range. However, since broadband was used for the network, there was a demand for making it faster and so that it would cover a wide area. The ring network is widely used for a wide area network (WAN) since it is easy to maintenance and readily avoids faults.

[0045] Figure 9 indicates a network wherein three rings 41 through 43 are interconnected by bridge A and bridge B. In Figure 9, we shall first of all explain operations between ring 41 and ring 42. Ring 41 is configured of bridge A, which is the contact point between bridges 1-1 through 1-4 and ring 42; and of bridge B, which is the contact point with ring 43. The respective bridges are connected on a 1:1 basis, thereby configuring a ring-shaped bridge network.

[0046] Ring 42 is configured of bridge A (port used separately from the port used in ring 41), which is the contact point with bridges 2-1 through 2-5 and ring 41. Each bridge is connected on a 1:1 basis, thereby configuring a ring-shaped bridge network.

[0047] Here, bridge A is configured of the bridge in the present invention. Bridge A is defined as having four ports and one of the spanning tree management parts is defined for the two ports which make up ring 41. The other two ports on bridge A make up ring 42. These two ports are such that the other spanning tree management part is defined by

/7

the network manager. As a result, the respective other spanning tree protocols operate inside rings 41 and 42. In Figure 9, each of the spanning trees 111, 112 is configured inside rings 41, 42, as indicated by the thick line in the figure.

[0048] Likewise, bridge B is configured by the bridge in the present invention between rings 41, 43. Here, the same operations are carried out as between rings 41, 42 so that still another spanning tree 113 is configured inside ring 43.

[0049] Thus, in this practical embodiment, the spanning topology is closed inside a certain network group (here, the ring). As a result, reconfiguration of another spanning tree does not affect the inside of another network group when a fault occurs inside a single network group. In

addition, service inside a network group which has not been reconfigured is not interrupted.

[0050] Furthermore, the flow of operations indicated in Figure 5 and Figure 6 is such that it is stored in advance as a program inside ROM and other read only storage media and needless to say, this can be accomplished by having this read and executed by the computer. In addition, we have explained the LAN as an applications network. However, it is not limited to this and it may be applied as well to general networks. As a result, the bridge indicated in Figure 1 can be applied widely to a communication device which comprises a data exchange function.

[0051]

[Effect of Invention] As explained previously, the present invention has the following effects. The first effect is that the total number of bridges used to configure a single spanning tree can be reduced by partitioning a spanning tree configured in the LAN and other network in the same segment. As a result, the reconfiguration of the spanning tree due to a fault at any location can be carried out more quickly.

[0052] The second effect is that since the spanning tree configured in the network is partitioned, the parameters of the fault are kept to a minimum and service can continued uninterrupted within the parameters of the spanning tree regardless of the tree reconfiguration.

[0053] The third effect is that the method of allocation of the plurality of ports and the plurality of spanning tree management parts is made so that they can be controlled with flexibility so that the spanning tree can be configured independent of each ring by connecting a plurality of ring networks.

[Brief Explanation of Figures]

[Figure 1] A block diagram of the bridge in a practical embodiment of the present invention.

[Figure 2] A diagram of an example of the port-to-spanning tree management part allocation table in Figure 1.

[Figure 3] A block diagram of an example of the frame processing part in Figure 1.

[Figure 4] A diagram indicating the format of the C-BPDU.

[Figure 5] A diagram indicating the schematic flow of operations in the present invention.

[Figure 6] A diagram indicating the flow of the details of the spanning tree buildup processing procedures.

[Figure 7] A configuration example of the network in a practical embodiment of the present invention.

[Figure 8] A diagram indicating an example of building the spanning tree in the network configuration in Figure 7.

[Figure 9] A diagram indicating an example of the configuration of the spanning tree in another practical embodiment of the present invention.

[Figure 10] An explanatory diagram of the prior-art technology.

[Explanation of Notation]

A through F..LAN

1 through 9..bridge

11 through 1n.. frame processing part

21..CPU

22..port-to-spanning tree management part allocation table

31 through 3m..spanning tree management part

51..frame receiving part

52..C-BPDU terminal part

53..frame transfer management part

54..port information table

55..frame sending part

56..C-BPDU generation part

57..port ID/port path cost table

58..bridge ID/route path cost table

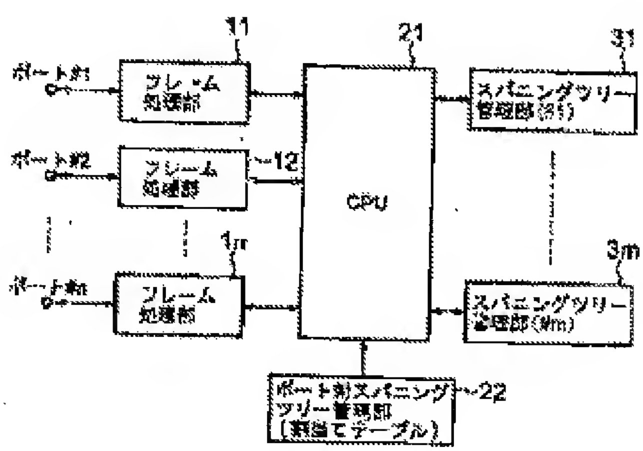
[Figure 2]

22

ポート#	スパニングツリー管理部#
1	1
2	3
3	3
⋮	⋮

Port # Spanning Tree Management Part (22)

[Figure 1]



[Captions:

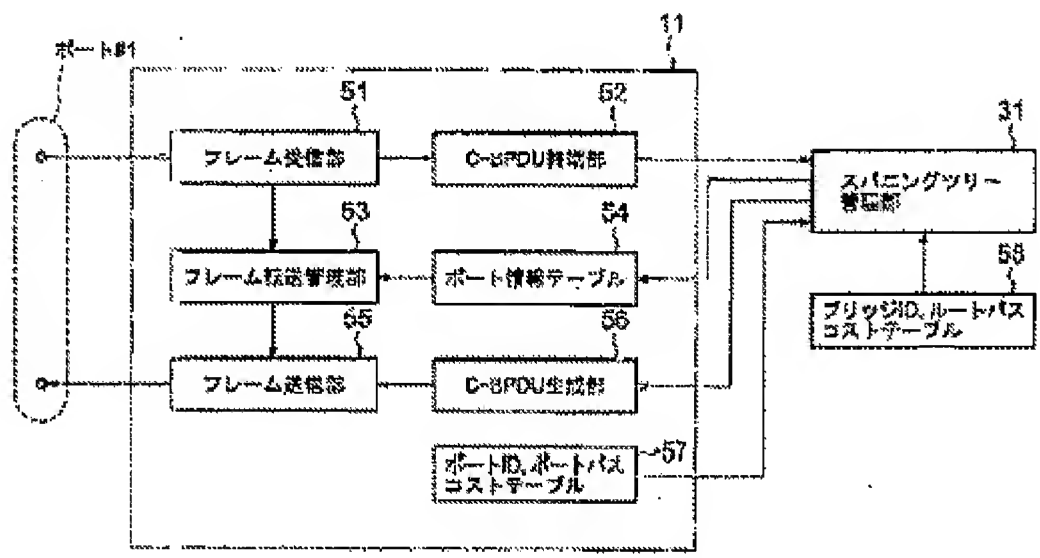
Port #1

Port #2

Port #3

11:frame processing part; 21: frame [illegible] part; 31: spanning tree management part (S1); 3m: spanning tree management part ([illegible]); 22: port:spanning tree management part (allocation table); 1m: frame processing part.

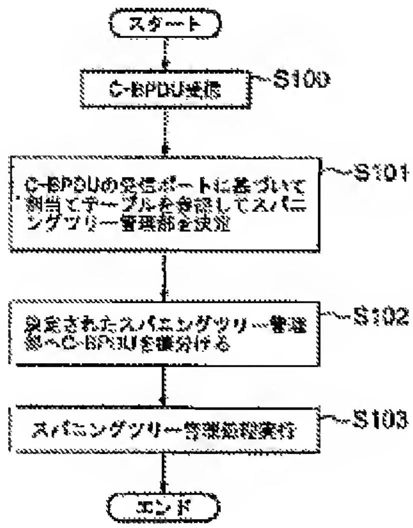
[Figure 3]



[captions:

Port #1; 51: frame receiving part; 53: frame [illegible] management part; 55: frame transfer part; 52: C-BPDU [illegible] part; 54: port information table; 56: C-BPDU generation part; 57: port ID, port path cost table; 31: spanning tree management part; 58: bridge ID, route path cost table;

[Figure 5]



[captions:

START

S100: C-BPDU received

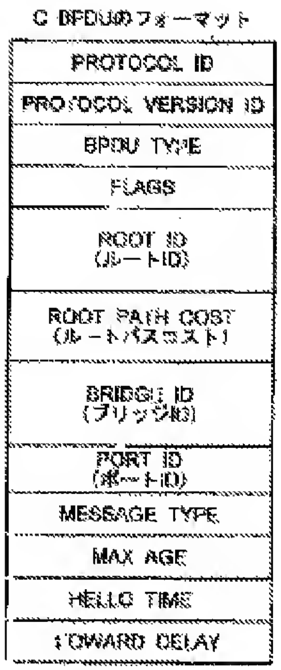
S101: [illegible] receive table [illegible] determine spanning tree management part based on [illegible] port of C-BPDU

S102: [illegible] C-BPDU for spanning tree management part determined

S103: execute [illegible] spanning tree

END

[Figure 4]



[Captions:

C-BPDU FORMAT

PROTOCOL ID

PROTOCOL VERSION ID

BPDU TYPE

FLAGS

ROOT ID

ROOT PATH COST

BRIDGE ID

PORT ID

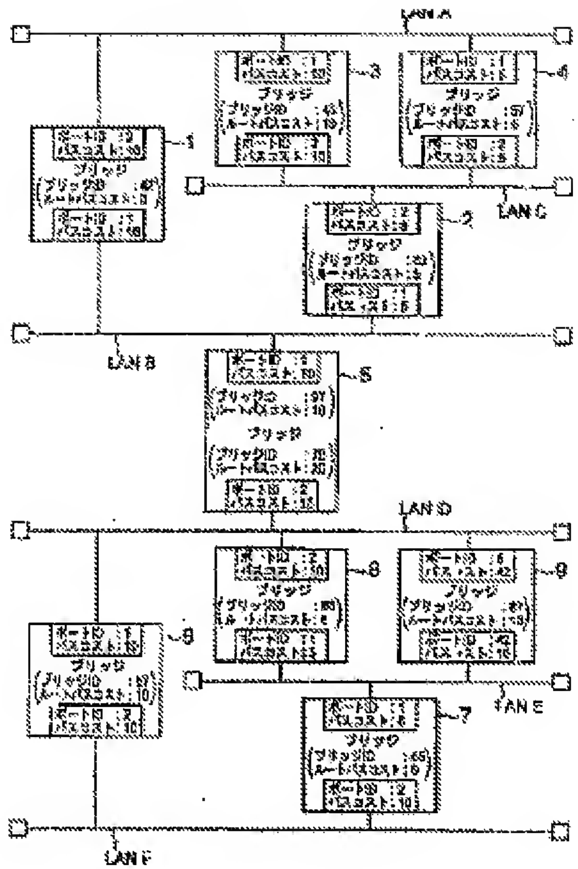
MESSAGE TYPE

MAX AGE

HELLO TIME

TOWARD DELAY

[Figure 7]



[Captions:

1: port ID: 2

Path cost 10

Bridge

Bridge ID: 42

Root path cost : 0

```
   Port ID: 1
   Path cost: 10
2: port ID: 2
   Path cost: 5
   Bridge
  (bridge ID: [illegible]
  Root path cost: 6
  Port ID: 1
  Path cost: 5
3: port ID: 1
   Path cost: 10
   Bridge
   (bridge ID: 46
   Root path cost: [illegible]
   Port ID: 2
   Path cost: 10
4: port ID: 1
   Path cost: 5
   Bridge
   (bridge ID: 57
   Root path cost: 6
   Port ID: 2
   Path cost: 5
5: port ID: [illegible]
```

Path cost: 10

Bridge

(bridge ID: 10

Root path cost: 70

Path cost: 20

6: port ID: 1

Path cost: 10

Bridge

(bridge ID: 10

Root path cost: 10

Port ID: 2

Path cost: 10

7: port ID: 1

Path cost: 6

Bridge

(bridge ID: [illegible]

Root path cost: 0

Port ID: 2

Path cost: 10

8: port ID: 2

Path cost: 10

Bridge

(bridge ID: [illegible]

Root path cost: [illegible]

Port ID: 1

Path cost: 5

9: port ID: 6

Path cost: 42

Bridge

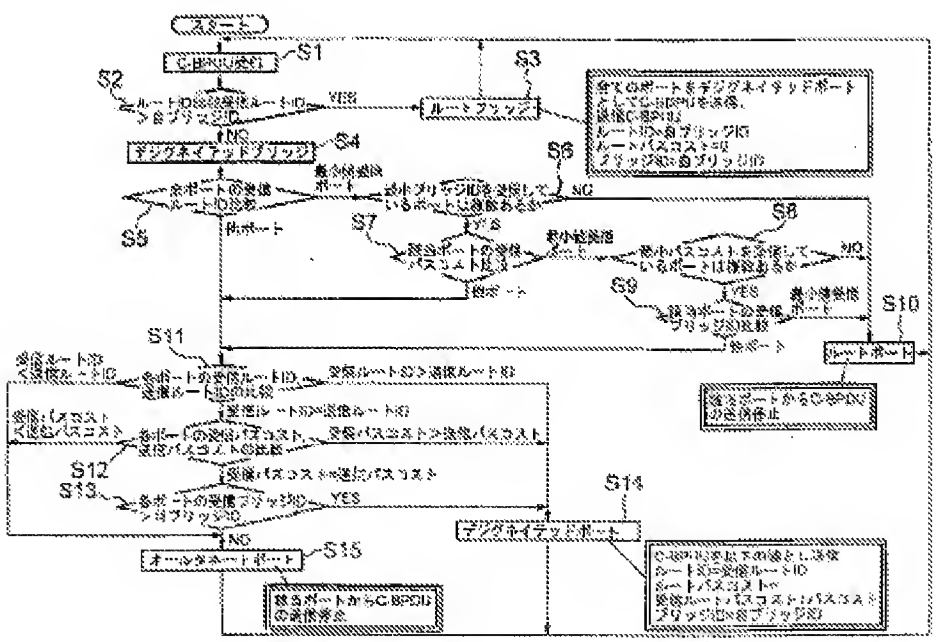
(bridge ID: [illegible]

Root path cost: 10

Port ID: 42

Path cost: 10

[Figure 6]

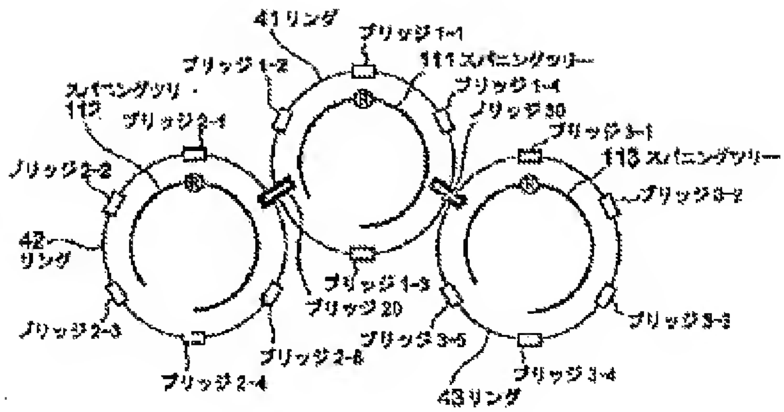


[captions:

1: Start; S1: C-BPDU received; S2: root ID comparison receive root ID, own bridge ID; S3: root bridge; S4: designated bridge; 2: minimum value receiving port; S5: all

ports received, root ID comparison; 3: other port; S6: more than one port where minimum bridge ID received?; S7: relevant port received, path cost comparison; 4: other port; 5: port where minimum value received; 6: send C-BPRU as designated port for all ports; send C-BPDU, root ID = self bridge ID, root path cost = 0; bridge ID = self bridge ID; S8: more than one port where minimum cost received?; S9: relevant port received, bridge ID comparison; 7: other port; 8: minimum value received port; S10: root port; S11: each port receiving root ID, comparison of sending root ID; 9: receive root ID <send root ID; 10: receive path cost, <send path cost; 11: receive root ID> send root ID; 12: receive root ID = send root ID; S12: each port's receive path cost, send path cost comparison; 13: receive path cost > send path cost; 14: path cost = send path cost; S13: receive bridge ID for each port > own bridge ID; S15: alternate port; 15: stop sending C-BPDU from relevant port; S14: designated port; 16: send C-BPDU as follow value; root ID = receive root ID; root path cost = receive root path cost : path cost bridge ID = own bridge ID

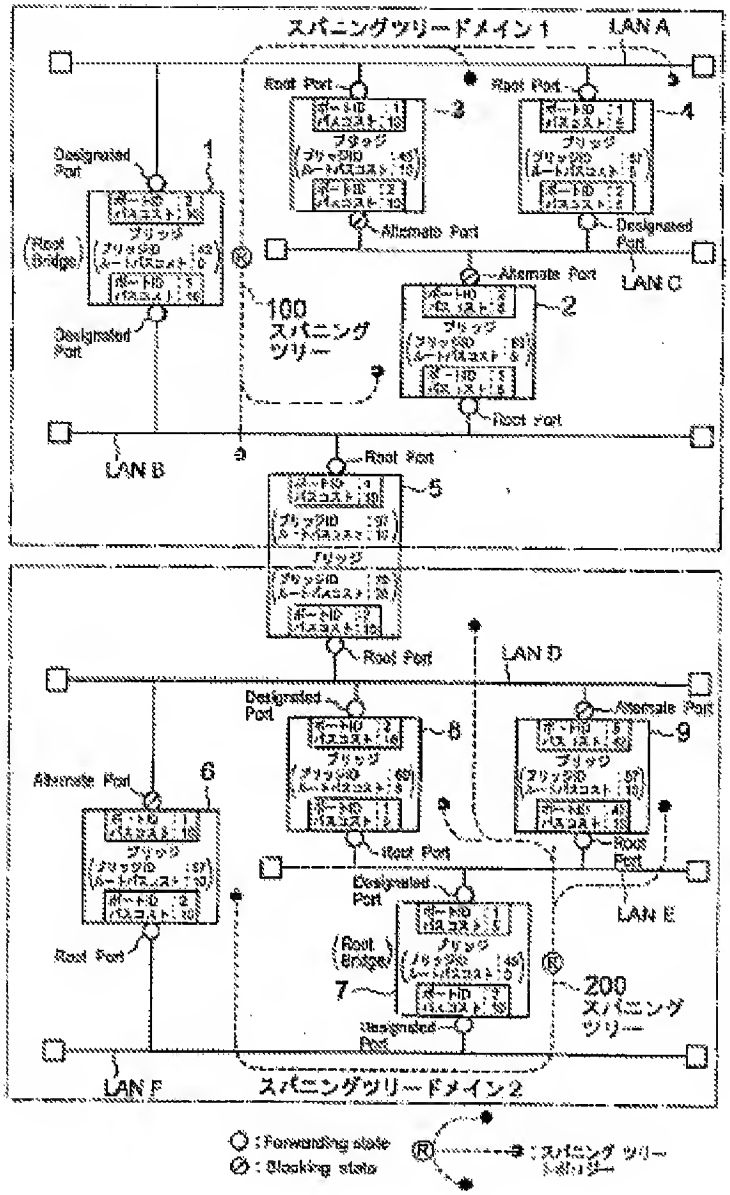
[Figure 9]



[captions:

41: ring; 1-2: bridge; 112: spanning tree; 2-1: bridge; 2-2: bridge; 42: ring; 2-3: bridge; 2-4: bridge; 1-1: bridge; 111: spanning tree; 1-4: bridge; 30: bridge; 3-1: bridge; 113: spanning tree; 3-2: bridge; 3-3: bridge; 3-4: bridge; 43: ring; 1-3: bridge; 20: bridge; 3-5 bridge

[Figure 8]



スパニングツリードメイン1
LAN A
Root Port
Designated Port
Alternate Port
LAN B
LAN C
LAN D
LAN E
LAN F
100 スパニングツリー
200 スパニングツリー
Root Bridge
スパニングツリードメイン2
○ : Forwarding state
⊘ : Blocking state

[Captions:

Top of figure: spanning tree domain 1

1: port ID: 2

Path cost: 10

Bridge

(bridge ID: 42

Root path cost: 0

Port ID: 1

Path cost: 10

2. port ID: 2

Path cost: 5

Bridge

(bridge ID: 83

Root path cost: 5

Port ID: 1

Path cost: 3

3. port ID: 1

Path cost: 10

Bridge

(bridge ID: 45

Root path cost: 10

Port ID: 2

Path cost: 10

4. port ID: 1

Path cost: 5

Bridge

(bridge ID: 57

Root path cost: 5

Port ID: 2

Path cost: 5

5.port ID: 1

Path cost: 10

Bridge

(bridge ID: 97

Root path cost: 10

Port ID: 70

Path cost: 20

6.port ID: 1

Path cost: 10

Bridge

(bridge ID: 57

Root path cost: 10

Port ID: 2

Path cost: 10

7.port ID: 1

Path cost: 5

Bridge

(bridge ID: 46

Root path cost: 0

Port ID: 2

Path cost: 10

8.port ID: 2

Path cost: 10

Bridge

(bridge ID: 60

Root path cost: 5

Port ID: 1

Path cost: 5

9.port ID: 5

Path cost: 42

Bridge

(bridge ID: 87

Root path cost: 10

Port ID: 42

Path cost: 10

100: Spanning Tree

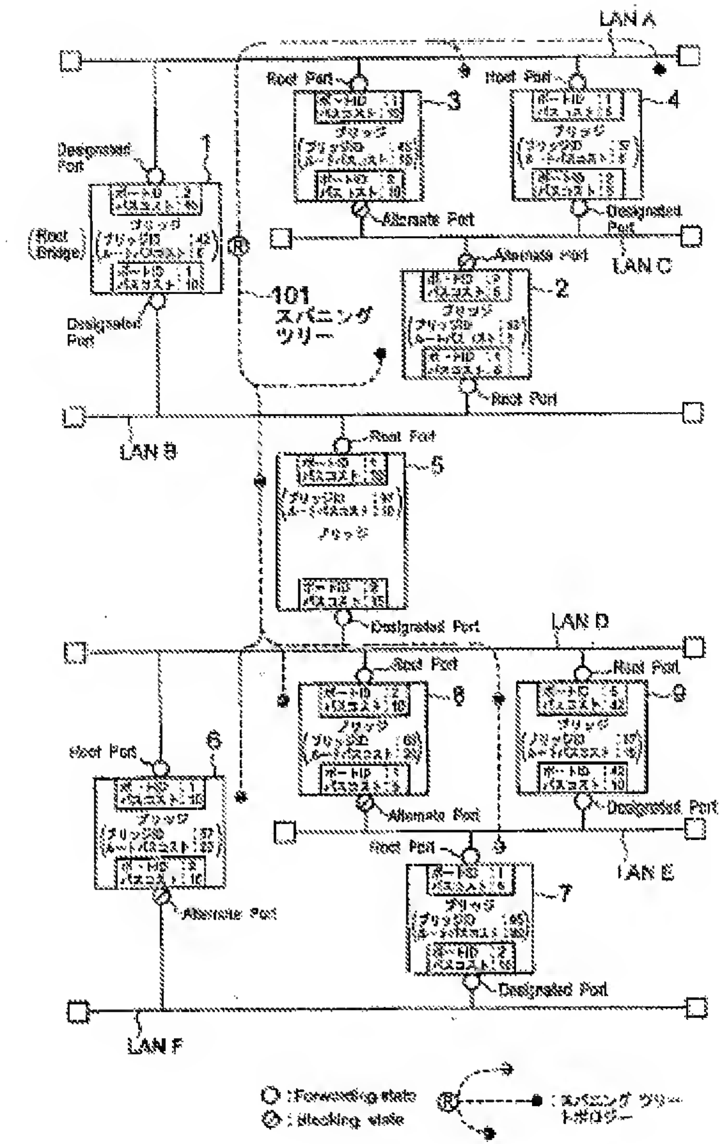
Bottom of figure: Spanning Tree Domain 2

0 forwarding spanning tree

0 blocking spanning tree

Spanning tree domain 2

[Figure 10]



[Captions:

1: port ID: 2

Path cost: 10

Bridge ID: 42

Root path cost: 0

Port ID: 1

Path cost: 10

2.port ID: [illegible]

Path cost: 5

Bridge

(bridge ID: [illegible]

Root path cost: [illegible]

Port ID: 1

Path cost: 5

3. port ID: 1

Path cost: 10

Bridge

(bridge ID: 45

Root path cost: 10

Port ID: 2

Path cost: 10

4. port ID: 1

Path cost: 5

Bridge

(bridge ID: 57

Root path cost: 2

Port ID: 2

Path cost: 5

5: port ID: 1

Path cost: 10

bridge

(Bridge ID: 97

Root path cost: 10

Port ID: 2

Path cost: 15

6.port ID: 1

Path cost: 10

Bridge

(bridge ID: 57

Root path cost: 20

Port ID: 2

Path cost: 10

7. port ID: 1

Path cost: 5

Bridge

(bridge ID: 45

Root path cost: 20

Port ID: 2

Path cost: 10

8.port ID: 2

Path cost: 10

Bridge

Bridge ID: 60

Root path cost: 20

Port ID: 1

Path cost: 5

9.port ID: 5

Path cost: 42

Bridge

(bridge ID: [illegible]

Root path cost: 15

Port ID: 42

Path cost: 10